

PREDICTING DRIVER'S INJURY SEVERITY IN
AUTOMOBILE HEAD-ON COLLISIONS
USING MACHINE LEARNING

By

MIAO MEI CHONG

Bachelor of Science

Guangzhou Medical College

Guangzhou, China

1987

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
August 2003

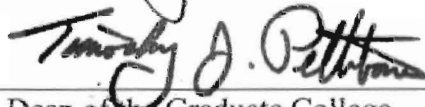
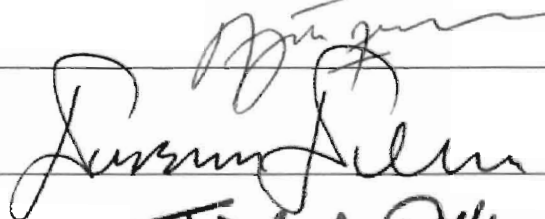
Oklahoma State University Library

PREDICTING DRIVER'S INJURY SEVERITY IN
AUTOMOBILE HEAD-ON COLLISIONS
USING MACHINE LEARNING

Thesis Approved:



Thesis Adviser



Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my appreciation to Dr. Marcin Paprzycki, Dr. Ajith Abraham, and Dr. Dursun Delen for serving as my committee members. I am grateful to my advisor, Dr. Paprzycki for his constant support, endless energy and drive. Dr. Delen helped me get started by providing me with the dataset and the initial information for my research. Dr. Abraham has provided me with his advice, time and expertise on the subject of machine learning. I further extend my thanks to Dr. Srinivas Mukkamala, professor in the Computer Science Department, New Mexico College of Mining and Technology for providing me with technical advice in support vector machines.

I am forever indebted to my family for their love and support, especially to my husband who always has confidence in me.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
II. RELATED WORK	3
III. AN OVERVIEW OF DATA MINING TECHNIQUES	7
3.1 Multilayer Perceptron Neural Networks	7
3.2 Decision Tree	9
3.3 Support Vector Machines	10
IV. PREPARATION FOR EXPERIMENT	13
4.1 Description of Dataset	13
4.2 Business Understanding	14
4.3 Data Understanding and Data Preparation	15
V. Modeling and Evaluation	19
5.1 Modeling With All Output Classes	19
5.2 Modeling With One Output Class At A Time	21
5.2.1 MLP Result	22
5.2.2 Decision Tree Result	25
5.2.3 SVM Result	35
5.3 Model Evaluation	37
VI. SUMMARY	38
SELECTED REFERENCES	39
APPENDICES	42
APPENDIX A: VARIABLE DEFINITIONS FOR GES DATASET	42
APPENDIX B: SAMPLE DATA FROM ORIGINAL DATASET	46
APPENDIX C: SAMPLE DATA OF HEAD-ON FRONT IMPACT DATASET	47

LIST OF TABLES

Table	Page
1. Driver Injury Severity Distribution.....	17
2. Data Coding of the Input Variables	20
3. Training and Testing Results of No Injury	23
4. Training and Testing Results of Possible Injury	24
5. Training and Testing Results of Non-incapacitating Injury	24
6. Training and Testing Results of Incapacitating Injury	24
7. Training and Testing Results of Fatal Injury	25
8. Classification Matrix and Accuracy of C&RT	26
9. Parameter Setting and Accuracy (%) of RBF SVM	36
10. Performance Evaluation.....	37

LIST OF FIGURES

Figure	Page
1. Three-layer back-propagation neural network.....	8
2. Linear separating hyperplanes for the separable case.....	12
3. Linear separating hyperplanes for the non-separable case	12
4. Lift Chart for No Injury	27
5. Lift Chart for Possible Injury.....	27
6. Lift Chart for Non-incapacitating Injury.....	28
7. Lift Chart for Incapacitating Injury.....	28
8. Lift Chart for Fatal Injury	29
9. No Injury Tree Structure.....	30
10. Possible Injury Tree Structure	31
11. Non-Incapacitating Injury Tree Structure.....	32
12. Incapacitating Injury Tree Structure	33
13. Fatal Injury Tree Structure.....	34

CHAPTER I

INTRODUCTION

Data mining is often defined as finding useful hidden information in a database, also known as knowledge discovery in databases (KDD). Digital data acquisition and storage technology have led to a huge amount of data kept in databases and data warehouses. The fast-growing tremendous amount of data has far exceeded our human ability for comprehension without powerful tools. Data mining tools perform data analysis and may uncover important data patterns. The information and knowledge gained can contribute to business strategies, decision supports, and scientific and medical research. Fayyad et al. define data mining as the application of specific algorithms for extracting patterns from data, and KDD as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [Fayyad].

The costs of fatalities and injuries due to traffic accident have a great impact on society. In recent years, researchers have paid increasing attention at determining the factors that significantly affect driver injury severity in traffic accidents. There are several approaches that researcher employed to study injury severity. These include neural network, nesting logic formulation, log-linear model, etc. This thesis research applies classification techniques in data mining to traffic accident dataset to build models that predict injury severity. Applying data mining techniques to study traffic accident data records can help find the characteristic of drivers' behavior, roadway condition and

weather condition that cause different injury severity when accidents happen. This can help decision makers improve traffic safety control policies.

CHAPTER II

RELATED WORK

Abdelwahab et al. studied the 1997 accident data for the Central Florida area [Abdelwahab]. The analysis focused on two-vehicle accidents that occurred at signalized intersections. The injury severity was divided into three classes: no injury, possible injury and disabling injury. They applied gamma statistic to ordinal variables and chi-squared test to nominal variables to determine the significance of the variables to injury severity, therefore, reduced the number of the variables for the model building. They compared the performance of MLP neural networks and Fuzzy ARTMAP neural network, and found that MLP neural networks classification accuracy is higher than Fuzzy ARTMAP neural network. Fuzzy ARTMAP is a clustering algorithm that maps a set of input vectors to a set of clusters. The Neural Network Toolbox from the MATLAB library was used to train and test the MLP. The MLP neural network with Levenberg-Marquardt algorithm as training algorithm had 65.6 and 60.4 percent classification accuracy for the training and testing phases, respectively. Fuzzy ARTMAP neural network had a classification accuracy of 56.1 percent.

Yang, et al. used neural network approach to detect safer driving patterns that have less chances of causing death and injury when a car crash occurs [Yang]. They performed the Cramer's V Coefficient test [Zembowicz] to identify significant variables that cause injury, therefore, reduced the dimensions of the data for the analysis. Then,

they applied data transformation method with a frequency-based scheme to transform categorical codes into numerical values. They used the Critical Analysis Reporting Environment (CARE) system, which was developed at the University of Alabama, trained a back propagation neural network using the 1997 Alabama interstate alcohol-related data, and further studied the weights on the trained network to obtain a set of controllable cause variables that are likely causing the injury crash. The target variable in their study has two classes: injury and non-injury, in which injury class includes fatalities. They found that by controlling a single variable (such as driving speed, or light conditions) they could reduce fatalities and injuries by up to 40%.

Omar, et al. used neural network to analyze vehicle crashworthiness [Omar]. They used the equation of motion of the dynamic system to define the inputs and outputs of the ANN, and used the crash data available from test results or finite element simulation to train an especially configured Hopfield, recurrent ANN. They found that the acceleration, velocity, and displacement curves predicted by the ANN are almost identical to those obtained from finite element simulations. They used an ANN to store the nonlinear dynamic characteristics of the vehicle structure, and proved the concept of using the ANN in crashworthiness analysis.

Sohn, et al. applied data fusion, ensemble and clustering to improve the accuracy of individual classifiers for two categories of severity (bodily injury and property damage) of road traffic accident [Sohn]. The individual classifiers used are neural network and decision tree. They applied a clustering algorithm to the dataset to divide the data into subsets of data, and then used each subset of data to train the classifiers. They

found that classification based on clustering works better if the variation in observations is relatively large as in Korean road traffic accident data.

Mussone, et al. used ANN to analyze vehicle accident that occurred at intersections in Milan, Italy [Mussone]. They chose feed-forward neural networks with a back-propagation learning paradigm. The model has 10 input nodes for eight variables (day of night, traffic flows circulating in the intersection, number of virtual conflict points, number of real conflict points, type of intersection, accident type, road surface condition, and weather conditions), 4 hidden nodes, and 1 output node. The output node was called accident index, which was calculated as the ratio between the number of accidents for a given intersection and the number of accidents at the most dangerous intersection. Results showed that the highest accident index for running over of pedestrian occurs at non-signalized intersections at night-time.

Dia et al. used 'real-world' data for developing a multi-layer feed-forward (MLF) neural network freeway incident detection model [Dia]. The model used speed, flow and occupancy data measured at dual stations, averaged across all lanes. They compared the performance of the neural network model and the incident detection model in operation on Melbourne's freeways. Results showed that neural network model could provide faster and more reliable incident detection over the model that was in operation on Melbourne's freeways. They also found that failure to provide speed data at a station could significantly deteriorate model performance within that section of the freeway.

Shankar, et al. applied a nested logic formulation for estimating accident severity likelihood conditioned on the occurrence of an accident [Shankar]. They found that there

is a greater probability of evident injury or disabling injury/fatality relative to no evident injury if at least one driver did not use a restraint system at the time of the accident.

Kim et al. developed a log-linear model to clarify the role of driver characteristics and behaviors in the causal sequence leading to more severe injuries. They found that driver behaviors of alcohol or drug use and lack of seat belt use greatly increase the odds of more severe crashes and injuries [Kim].

Yeo, et al. considered the problem of predicting claim costs in the automobile insurance industry. They found that a data-driven clustering approach to risk classification could yield better quality predictions of expected claim costs compared to a heuristic approach [Yeo].

CHAPTER III

AN OVERVIEW OF DATA MINING TECHNIQUES

3.1 Multilayer Perceptron Neural Networks

Artificial neural network (ANN) is a computing technology that mimics certain processing capabilities of the human brain. An ANN consists of a number of interconnected neurons. The neurons are connected by weighted links passing signals from one neuron to another. Every neuron consists of a processing element with synaptic input connections and a single output. Haykin defines neural network as [Haykin]:

A massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects: (1) Knowledge is acquired by the network through a learning process, and (2) Interneuron connection strengths known as synaptic weights are used to store the knowledge.

A multilayer perceptron is a feed forward neural network with one or more hidden layers. The network consists of an input layer of source neurons, at least one hidden layer of computational neurons, and an output layer of computational neurons. The input layer accepts input signals and redistributes these signals to all neurons in the hidden layer. The output layer accepts a stimulus pattern from the hidden layer and establishes the output pattern of the entire networks [Negnevitsky]. Figure 1 is a three-layer back-propagation neural network. In this figure, the input layer has n neurons, the hidden layer has m neurons, and the output layer has l neurons. The connections in the MLP are allowed

from one layer to the next layer, no connections are allowed among the neurons belonging to the same layer.

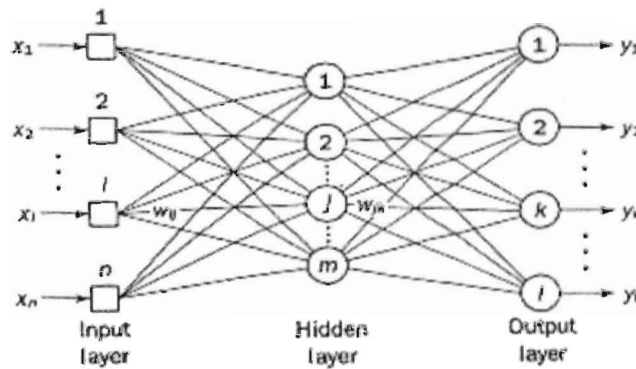


Figure 1: Three-layer back-propagation neural network

The MLP neural networks training phase works as follows: given a collection of training data $\{x_1(p), d_1(p)\}, \dots, \{x_i(p), d_i(p)\}, \dots, \{x_n(p), d_n(p)\}$, the objective is to obtain a set of weights that makes almost all the tuples in the training data classified correctly, or in other words, is to map $\{x_1(p) \text{ to } d_1(p)\}, \dots, \{x_i(p) \text{ to } d_i(p)\}$, and eventually $\{x_n(p) \text{ to } d_n(p)\}$. The algorithm starts with initializing all the weights (w) and threshold (θ) levels of the network to small random numbers. Then calculate the actual output of the neurons in the hidden layer as:

$$y_i(p) = f[\sum_{(i=1 \text{ to } n)} x_i(p) * w_{ij}(p) - \theta_j], \text{ where } n \text{ is the number of inputs of neuron } j \text{ in}$$

the hidden layer. Next calculate the actual outputs of the neurons in the output layer as:

$$y_k(p) = f[\sum_{(j=1 \text{ to } m)} x_{jk}(p) * w_{jk}(p) - \theta_k], \text{ where } m \text{ is the number of inputs of neuron}$$

k in the output layer. The weight training is to update the weights in the back-propagation network propagating backward the errors associated with output neurons. The error function is:

$$E(w) = \sum_{(p=1 \text{ to } PT)} \sum_{(i=1 \text{ to } l)} [d_i(p) - y_i(p)]^2, \text{ where}$$

$E(w)$ = error function to be minimized,

w = weight vector,

PT = number of training patterns,

l = number of output neurons,

$d_i(p)$ = desired output of neuron i when pattern p is introduced to the MLP, and

$y_i(p)$ = actual output of the neuron i when pattern p is introduced to the MLP. The

objective of weight training is to change the weight vector w so that the error function is minimized. By minimizing the error function, the actual output is driven closer to the desired output.

3.2 Decision Trees

Decision tree CART is one well-known algorithm for classification problems. The CART tree model consists of a hierarchy of univariate binary decisions [Hand]. Each internal node in the tree specifies a binary test on a single variable, branch represents an outcome of the test, each leaf node represent class labels or class distribution. CART operates by choosing the best variable for splitting the data into two groups at the root node, partitioning the data into two disjoint branches in such a way that the class labels in each branch are as homogeneous as possible, and splitting is recursively applied to each branch, and so forth.

If a dataset T contains examples from n classes, gini index, $gini(T)$ is defined as:

$gini(T) = 1 - \sum_{j=1}^n p_j^2$, where p_j is the relative frequency of class j in T [Han]. If

dataset T is split into two subsets T_1 and T_2 with sizes N_1 and N_2 , the gini index of the split data contains examples from n classes, the gini index $gini(T)$ is defined as:

$$gini_{\text{split}}(T) = N_1/N gini(T_1) + N_2/N gini(T_2).$$

CART exhaustively searches for univariate splits. The attribute provides the smallest $gini_{split}(T)$ is chosen to split the node. CART recursively expands the tree from a root node, and then gradually prunes back the large tree. The advantage of a decision tree is extracting classification rules from trees and is straightforward. It can represent the knowledge in the form of IF_THEN rules; one rule is created for each path from the root to a leaf node.

3.3 Support Vector Machines

Support Vector Machines (SVM) is an approach to machines learning based on statistical learning theory. SVMs have been successfully applied to a number of applications ranging from handwriting recognition, intrusion detection in computer networks, and text categorization to image classification, breast cancer diagnosis and prognosis and bioinformatics [Pal]. SVM has two key techniques, one is the mathematical programming and the other one is kernel functions. The parameters are found by solving a quadratic programming problem with linear equality and inequality constraints; rather than by solving a non-convex, unconstrained optimization problem. SVMs are kernel-based learning algorithms in which only a fraction of the training examples are used in the solution (these are called the Support Vectors), and where the objective of learning is to maximize a margin around the decision surface. The flexibility of kernel functions allows the SVM to search a wide variety of hypothesis spaces. The basic idea of applying SVMs to pattern classification can be stated briefly as: first map the input vectors into one feature space (possible with a higher dimension), either linearly or nonlinearly, which is relevant with the selection of the kernel function; then within the feature space, seek an optimized linear division, i.e. construct a hyperplane which

separates two classes. Campbell said "SVMs are the most well known of a class of algorithms which use the idea of kernel substitution" [Campbell].

For a set of n training examples (x_i, y_i) , where $x_i \in \mathbf{R}^d$ and $y_i \in \{-1, +1\}$, suppose there is a hyperplane, which separates the positive from the negative examples. The points x which lie on the hyperplane (H_0) satisfy $w \cdot x + b = 0$, the algorithm finds this hyperplane (H_0) and other two hyperplanes (H_1, H_2) parallel and equidistant to H_0 ,

$H_1: w \cdot x_i + b = 1, H_2: w \cdot x_i + b = -1, H_1$ and H_2 are parallel and no training points fall between them. Support vector algorithm looks for the separating hyperplane and maximizes the distance between H_1 and H_2 . So there will be some positive examples on H_1 and some negative examples on H_2 . These examples are called support vectors. The distance between H_1 and H_2 is $2/\|w\|$, in order to maximize the distance, we should minimize $\|w\| = w^T w$, subject to constraints $y_i (w \cdot x_i + b) \geq 1, \forall_i$

Introducing Lagrangian multipliers $\alpha_1, \alpha_2, \dots, \alpha_n \geq 0$, the learning task becomes

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^n \alpha_i [y_i (w \cdot x_i + b) - 1]$$

The above equation is for two classes that are linearly separable. When the two classes are non-linearly separable, SVM can transform the data points to another high dimensional space. Detailed description to the theory of SVMs for pattern recognition can be found in [Cristianini, Burges]. Figure 2 and 3 from [Burges] show linear separating hyperplanes for separable and non-separable case. Training of SVM involves optimization of a convex cost function, so there are no local minima to complicate the learning. Cristianini said "the four problems of efficiency of training, efficiency of testing, over fitting and algorithm parameter tuning are all avoided in the SVM design" [Cristianini].

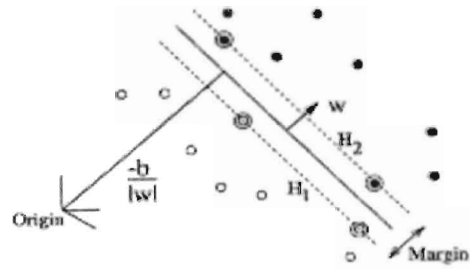


Figure 2: Linear separating hyperplanes for the separable case.

The support vectors are circled.

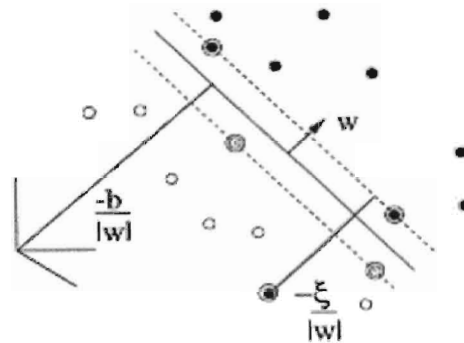


Figure 3: Linear separating hyperplanes for the non-separable case.

CHAPTER IV

PREPARATION FOR EXPERIMENT

CRISP-DM (Cross-Industry Standard Process for Data Mining) described in [Shearer] is used for the thesis study. CRISP-DM organizes the data mining process into six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment.

4.1. Description of Dataset

The dataset for the thesis study contains traffic accident records from 1995 to 2000, a total number of 417,670 cases. The dataset is obtained from GES (General Estimates System). GES provides traffic accident records that are free for the general public to use. Interested reader can go to GES website to find out more about GES data [GES]. According to the variable definitions for GES dataset, this dataset has drivers' only records and doesn't include passengers' information. It includes labels of year, month, region, primary sampling unit, the number of the police jurisdiction, case number, person number, vehicle number, vehicle make and vehicle model; inputs of age, gender, alcohol, rest system, eject, body type, vehicle age, vehicle role, initial point of impact, manner of collision, rollover, roadway surface condition, light condition, travel speed, and speed limit; output is injury severity. Appendix A contains precise definitions of variables occurring in the dataset. The injury severity has five classes: No Injury,

Possible Injury, Non-incapacitating Injury, Incapacitating Injury, and Fatal Injury. In the original dataset, 70.18% of the cases have output of no injury, 16.07% of the cases have output of possible injury, 9.48% of the cases have output of non-incapacitating injury, 4.02% of the cases have output of incapacitating injury, and 0.25% of the cases have fatal injury. Appendix B shows sample records of the original dataset.

4.2 Business Understanding

The business-understanding phase involves determining business objectives, assessing the situation, determining the data mining goals, and producing the project plan [Shearer].

Engineers and researchers in automobile industry have tried to design and build safer automobile, but traffic accidents are unavoidable due to various factors. The pattern of dangerous crash can be detected if we develop a prediction model that automatically predicts injury severity of traffic accidents. These behavioral and roadway patterns are useful in development of traffic safety control policy.

The records in the dataset are input/output pairs; each record has a known output. The output variable-injury severity is categorical, and has five classes. A supervised learning algorithm will try to map an input vector to the desired output class. Classification predictive model will serve our problem domain. After a model is trained, it will be used to predict future cases. This research used neural network, decision tree, and support vector machines to explore the performance of each algorithm, and find the best model for the prediction.

4.3 Data Understanding and Data Preparation

This step examine the properties and format of the data, explore the data, verify the data quality, and then select, clean, and format the data [Shearer]. This prepares a dataset for model training and testing.

Labeled variables are mainly for identifying the cases. The variables that are irrelevant to the mining task are left out. The input and output variables are considered for the model building. There are no conflicts between the attributes since each variable represents its own characteristic. The variables are already categorized and are represented by numbers.

The manner of collision has 7 categories: not collision, rear-end, head-on, rear-to-rear, angle, sideswipe same direction, and sideswipe opposite direction. The distribution for fatal injury is as follow: 0.56% for not collision, 0.08% for rear-end collision, 1.54% for head-on collision, 0.00% for rear-to-rear collision, 0.20% for angle collision, 0.08% for sideswipe same direction collision, 0.49% for sideswipe opposite direction collision. Since head-on collision has the highest percent of fatal injury; therefore, the dataset is narrowed down to head-on collision only. Head-on collision has 10,386 records. There are 160 records of head-on collision with fatal injury; all of these 160 records have impact point of front.

The initial point of impact has 9 categories: no damage/non-collision, front, right side, left side, back, front right corner, front left corner, back right corner, back left corner. A variable with too many categories will confuse the model during learning stage, to avoid confusion of the model learning, the initial point of impact will be focus on front impact only. The head-on collision with front impact has 10,251 records; this is 98.70%

of the 10,386 head-on collision records. Travel speed and speed limit will not be used in the model because there are too many of unknowns, 67.68% of the records' travel speed is unknown. The input variables are: age, gender, alcohol, rest system, eject, vehicle body type, vehicle role, vehicle age, rollover, road surface condition, light condition. Appendix C shows a sample of the data records in this dataset.

In the dataset (10251 records) of head-on and front impact, there are 5,173 (50.46%) records with no injury, 2138 (20.86%) records with possible injury, 1723 (16.81%) records with non-incapacitating injury, 1057 (10.31%) records with incapacitating injury, 160 (1.56%) records with fatal injury. Table 1 shows the summary of driver injury severity distribution for head-on collision and front impact point dataset. From Table 1, we can see that alcohol usage, not using seat belt, ejection of driver, driver's age that is 65 and older, vehicle rollover, and lighting condition caused higher percentages of fatal injury, incapacitating injury and non-incapacitating injury.

Table 1: Driver Injury Severity Distribution

Factor	No Injury	Pos injury	Non-incapacitating	Incapacitating	Fatal	Total
Age						
0 (24&under)	1629(52.80%)	608(19.71%)	505(16.37%)	307(9.95%)	36(1.17%)	3085
1 (25-64)	3171(49.88%)	1362(21.43%)	1075(16.91%)	654(10.29%)	95(1.49%)	6357
2 (65+)	373(46.11%)	168(20.77%)	143(17.68%)	96(11.87%)	29(3.58%)	809
Gender						
0 (Female)	1749(41.95%)	1072(25.71%)	778(18.66%)	507(12.16%)	63(1.51%)	4169
1 (Male)	3424(56.30%)	1066(17.53%)	945(15.54%)	550(9.04%)	97(1.59%)	6082
Eject						
0 (NoEject)	5171(50.55%)	2137(20.89%)	1719(16.80%)	1047(10.23%)	156(1.52%)	10230
1 (Eject)	2(9.52%)	1(4.76%)	4(19.05%)	10(47.62%)	4(19.05%)	21
Alcohol						
0 (NoAlcohol)	4997(51.35%)	2067(21.24%)	1600(16.44%)	935(9.61%)	133(1.37%)	9732
1 (Alcohol)	176(33.91%)	71(13.68%)	123(23.70%)	122(23.51%)	27(5.20%)	519
Rest_Sys						
0 (NoUsed)	337(27.44%)	193(15.72%)	336(27.36%)	283(23.05%)	79(6.43%)	1228
1 (Used)	4836(53.60%)	1945(21.56%)	1387(15.37%)	774(8.58%)	81(0.90%)	9023
Body_Typ						
0 (cars)	3408(47.49%)	1600(22.30%)	1272(17.73%)	780(10.87%)	116(1.62%)	7176
1 (Suv&Van)	747(56.59%)	259(19.62%)	189(14.32%)	111(8.41%)	14(1.06%)	1320
2 (Truck)	1018(58.01%)	279(15.90%)	262(14.93%)	166(9.46%)	30(1.71%)	1755
Veh_Role						
1 (Striking)	4742(49.86%)	2011(21.15%)	1636(17.20%)	970(10.20%)	151(1.59%)	9510
2 (Struck)	261(72.70%)	54(15.04%)	29(8.08%)	15(4.18%)	0(0%)	359
3 (Both)	170(44.50%)	73(19.11%)	58(15.18%)	72(18.85%)	9(2.36%)	382
Rollover						
0 (Norollover)	5069(50.78%)	2123(20.85%)	1699(16.69%)	1037(10.19%)	152(1.49%)	10180
1 (Rollover)	4(5.63%)	15(21.13%)	24(33.80%)	20(28.17%)	8(11.27%)	71
Sur_cond						
0 (Dry)	3467(49.97%)	1404(20.24%)	1190(17.15%)	750(10.81%)	127(1.83%)	6938
1 (Slippery)	1706(51.49%)	734(22.16%)	533 (16.09%)	307(9.27%)	33(1.00%)	3313
Light_cond						
0 (Daylight)	3613(51.18%)	1487(21.06%)	1174(16.63%)	688(9.75%)	98(1.39%)	7060
1(partialdark)	1139(52.71%)	465(21.52%)	348(16.10%)	186(8.61%)	23(1.06%)	2161
2 (Dark)	421(40.87%)	186(18.06%)	201(19.51%)	183(17.77%)	39(3.79%)	1030

According to Han, leaving out the relevant attributes and keeping the irrelevant attributes may cause confusion for the mining algorithm employed [Han]. We can apply the attribute subset selection techniques to find a minimum set of attributes so that the resulting probability distribution of the data classes is as close as possible to the original

distribution of all attributes. To determine the “best” and “worst” attributes, we can use tests of statistical significance, such as chi-squared (χ^2) testing, which assume that the attributes are independent of one another. The input variable age, gender, alcohol, rest system, eject, vehicle body type, vehicle role, rollover, road surface condition, light condition are categorical variables. A chi-squared (χ^2) test is applied to test the dependence of input and output variables. The χ^2 test indicated that all these variables are significant (p -value < 0.05), so all of these variables will be used for modeling.

CHAPTER V

Modeling and Evaluation

In this phase, various modeling techniques are selected and applied. Modeling steps include the selection of the modeling technique, the generation of the test design, the creation of the models, and the assessment of the models [Shearer].

5.1 Modeling With All Output Classes

Webstatistica is a web base statistic and data mining tool. It allows user to train a neural network in two phases, each phase has several algorithms available. The available algorithms are Back-propagation, Conjugate gradient descent, Quasi-Newton, Levenberg-Marquardt, Quick Propagation, Delta-bar-delta. MLP on Webstatistica is used for the model building. MLP on Webstatistica will automatically divide dataset into training, cross-validation, and testing sets. As the neural network is trained, the software also does cross-validation; it provides an estimate of generalization performance. Using cross-validation will prevent model over training. As Bigus said: "If the same training patterns or examples are given to the neural network over and over, and the weights are adjusted to match the desired outputs, we are essentially telling the network to memorize the patterns, rather than to extract the essence of the relationships [Bigus]." An over trained neural network model cannot generalize and does not perform well on new cases. On Webstatistica, once the model finish the given algorithm and number of epochs of

training, it will give the result of the model at the best stopping point for training, indicate how many epochs have trained, and output the training, cross-validation, and testing performances and error rates.

First we used the dataset with only one output column that has all of the 5 output classes, the number of 0, 1, 2, 3, 4 represent no injury, possible injury, non-incapacitating injury, incapacitating injury and fatal injury respectively. Because the unbalancing number of records in each injury level, the use of the whole dataset to train a MLP will have 95% accuracy for no injury, and the rest of the injury class will get close to 0 accuracy. In order to train a model with no bias towards any injury class, a sample dataset with approximately equal amount of records in all injury class is needed. This sample dataset is obtained by running the Stratified Random Sampling algorithm on Webstatistica. This random sampling dataset has 814 records, with 173 records in no injury, 173 records in possible injury, 150 records in non-incapacitating injury, 158 records in incapacitating injury, and 160 records in fatal injury. Webstatistica used two-state or one-of-N coding for categorical variables, and used linear shift and scale for numerical variables. The coding for the input variables is described in Table 2.

Table 2: Data Coding of the Input Variables

Factor	Input Coding/Number of neurons
Age	One-of-N / 3
Gender	Two-state / 1
Alcohol	Two-state / 1
Rest_sys	Two-state / 1
Eject	Two-state / 1
Body_Type	One-of-N / 3
Veh_Age	Numerical value / 1
Veh_Role	One-of-N / 3
Rollover	Two-state / 1
Sur_Cond	Two-state / 1
Light_Cond	One-of-N / 3

Using the sampling dataset of 814 records, MLP with one hidden layer was used, and experimented with different number of hidden neurons. A MLP with 12 hidden neurons, used back-propagation for the first phase training with 100 epochs, a learning rate of 0.01, and conjugate gradient descent for second phase training with 500 epochs. It gives a result of accuracy of 56.3%, 14.6%, 19.3%, 49.7%, and 37.0% for fatal injury, incapacitating injury, non-incapacitating injury, possible injury, and no injury respectively. The overall classification accuracy is 35.87%.

Using the same sampling dataset, the advanced C&RT on Webstatistica was used with Gini goodness of fit measure, and estimated prior class probabilities, misclassification error as the stopping option for pruning, 10_fold cross-validation. It gives a result of accuracy of 75%, 4.67%, 0%, 61.85%, and 50.87% for fatal injury, incapacitating injury, non-incapacitating injury, possible injury and no injury respectively. The overall classification accuracy is 39.56%. These results show that the performance is very poor. Further data preparation is needed in order to help a model learn the data patterns.

5.2 Modeling With One Output Class At A Time

Training all five classes together, the model performance was not good. Instead of training five classes all at once, we separately trained one class at a time. Separated each output class used one-against-all approach. This approach selects one output class to be the positive class, and all the other classes to be the negative class. We set the positive class output to 1, and the negative class to 0 for neural network and decision tree. For SVMs we followed the data format requirement of the software.

5.2.1 MLP Result

We used a MLP with one hidden layer. Abraham said “a much used approximation for the number of hidden neurons for a three layer network is $N = \frac{1}{2} (J + K) + \sqrt{P}$, where J and K are the number of input and output neurons and P is the number of patterns in the training set” [Abraham]. We started with hidden neuron number = $\frac{1}{2} (J + K) + \sqrt{P}$. So we first used hidden neuron number = 95, and experimented with different values (in increment or decrement of 5). The number of hidden neurons that gives the best network performance will be selected for that class.

We used a combination of Back-propagation (BP) and Conjugate gradient descent (CG), and Levenberg-Marquardt (LM), for phase I and Phase II training. Webstatistica used hyperbolic activation function, $(e^x - e^{-x}) / (e^x + e^{-x})$ in the hidden layer, and logistic activation function, $1 / (1 + e^{-x})$ in the output layer. We train MLP on Webstatistica with 11 input variables; there will be 19 input neurons based on the coding listed on Table 2. For each output class, we experimented with different number of hidden neuron and different combination of algorithms for phase I and phase II. Our experiments showed, if we selected back-propagation to be the algorithm for first phase, then the software will automatically choose conjugate gradient descent to be the algorithm for second phase, no matter which algorithm we chose for the second phase. We trained models with back-propagation (100epochs, learning rate 0.01) and conjugate gradient descent (500 epochs). We also trained models with Levenberg-Marquardt (100 epochs, learning rate 0.01), used sum-squared error function for both methods of training, and compared model's performance. Tables 3 to 7 show the results of each model. From these tables we can see the best model for no injury class is LM with 45 hidden neurons,

its performance is 60.5% for testing; the most important variables are seat belt usage, gender, and body type of the vehicle. For possible injury class, the best model is BP-CG with 65 hidden neurons, its performance is 57.58% for testing; the most important variables are body type of the vehicle, driver's age, and light condition of the roadway. For non-incapacitating injury class, the best model is BP-CG with 75 hidden neurons, its performance is 56.8% for testing; the most important variables are driver's age, light condition of the roadway, the body type of the vehicle. For incapacitating injury class, the best model is LM with 40 hidden neurons, its performance is 63.43% for testing; the most important variables are seat belt usage, body type of the vehicle, driver's alcohol usage. For fatal injury class, the best model is BP-CG with 45 hidden neurons, its performance is 75.17% for testing; the most important variables are seat belt usage, light condition of the roadway, driver's alcohol usage.

Table 3: Training and Testing Results of No Injury

Injury Level	Model	# hidden Neuron	Train (%)	Test (%)	Note
No Injury	BP-CG	60	63.57	59.67	
		65	63.86	60.45	
		70	63.93	60.25	
		75	64.38	57.43	
		80	63.64	58.89	
	LM	35	63.8	58.02	
		40	64.46	58.94	
		45	62	60.5	
		50	64.23	59.52	

Table 4: Training and Testing Results of Possible Injury

Injury Level	Model	# hidden Neuron	Train (%)	Test (%)	Note
Possible Injury	BP-CG	65	59.34	57.58	
		70	59.56	55.15	
		75	58.88	57.29	
		80	58.39	56.22	
		95	60.07	55.93	
		100	61.48	57.14	
	LM	35	60.28	55.59	
		40	21.6	21.67	(classified all cases to positive cases)
		45	59.62	52.14	
		50	20.68	21.42	(classified all cases to positive cases)

Table 5: Training and Testing Results of Non-incapacitating Injury

Injury Level	Model	# hidden Neuron	Train (%)	Test (%)	Note
Non-incapacitating	BP-CG	60	57.88	55.25	
		65	57.69	54.66	
		75	58.71	56.8	
		80	57.78	54.13	
		85	57.83	55.59	
		90	60.36	55	
	LM	35	58.94	56.51	
		40	48.4	50.34	
		45	82.8	84.01	(classified all cases to negative class)
		50	61.39	56.41	

Table 6: Training and Testing Results of Incapacitating Injury

Injury Level	Model	# hidden Neuron	Train (%)	Test (%)	Note
Incapacitating	BP-CG	60	63.4	63.36	
		65	62.23	61.32	
		75	61.06	61.52	
		84	63.23	58.41	
		90	59.32	59.08	
	LM	35	54.81	54.62	
		40	62.46	63.46	
		45	59.31	58.45	
		50	89.5	89.26	(classified all cases to negative class)

Table 7: Training and Testing Results of Fatal Injury

Injury Level	Model	# hidden Neuron	Train (%)	Test (%)	Note
Fatal Injury	BP-CG	45	77.26	75.17	
		57	74.78	70.65	
		65	69.81	69.73	
		75	60.19	59.62	
		80	74.33	71.77	
	LM	35	55.97	55.05	
		40	0.02	0.02	(classified all cases to positive class)
		45	0.02	0.02	(classified all cases to positive class)
		50	58.14	60.2	

5.2.2 Decision Tree Result

We used the advanced classification tree (C&RT) on Webstatistica for our decision tree models. We trained each class with Gini goodness of fit measure, the prior class probabilities was set to equal, the stopping option for pruning was misclassification error, the minimum n per node was set to 5, fraction of objects was 0.05, the maximum number of nodes was 1000, the maximum number of level in tree was 32, the number of surrogates was 5, we used 10 fold cross-validation, and generated the comprehensive results. C&RT built the tree as user asked; then pruned the tree to smaller size. If pruning caused higher misclassification error, the model stopped pruning. When the model finished, it outputted the tree structure in a graph and in the table, classification matrix graph and table, predicted value, results of terminal nodes, and a lift chart.

The no injury model gives 62.46% accuracy for no injury class. The possible injury model gives 82.48% accuracy for possible injury class. The non-incapacitating injury model gives 85.70% accuracy for non-incapacitating injury class. The incapacitating injury model gives 77.63% for incapacitating injury class. The fatal injury model gives 100% accuracy for fatal injury class. Table 8 shows the classification matrix

and accuracy for all the models. Figure 4 to 8 show the lift chart for each model. In a lift chart, the area under the curve represents the accuracy of the model, the bigger the area under the curve, the higher the accuracy of the model. Figure 9 to 13 show the tree structure for each class's model. We also experimented with other available goodness of fit options and other options for stopping pruning, but its performance is not as good.

Table 8: Classification Matrix and Accuracy

Classification matrix			
Response: No injury			
	0	1	overall: 68.16%
0	3019	1558	class 0: 73.96%
1	1063	2592	class 1: 62.46%
Classification matrix			
Response: Possible Injury			
	0	1	overall: 66.28%
0	4030	303	class 0: 61.97%
1	2473	1426	class 1: 82.48%
Classification matrix			
Response: Non-Incapacitating			
	0	1	overall: 66.16%
0	4265	197	class 0: 62.23%
1	2589	1181	class 1: 85.70%
Classification matrix			
Response: Incapacitating			
	0	1	overall: 72.61%
0	5321	189	class 0: 72.03%
1	2066	656	class 1: 77.63%
Classification matrix			
Response: Fatal			
	0	1	overall 91.61%
0	7411		class 0: 91.47%
1	691	130	class 1: 100.00%

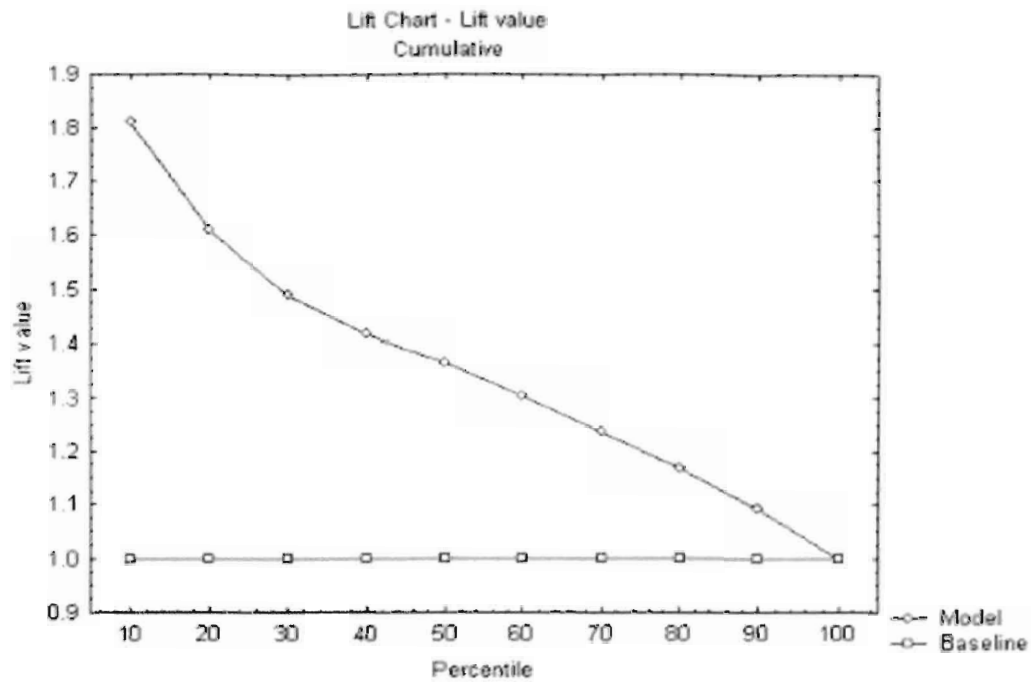


Figure 4: Lift Chart for No Injury

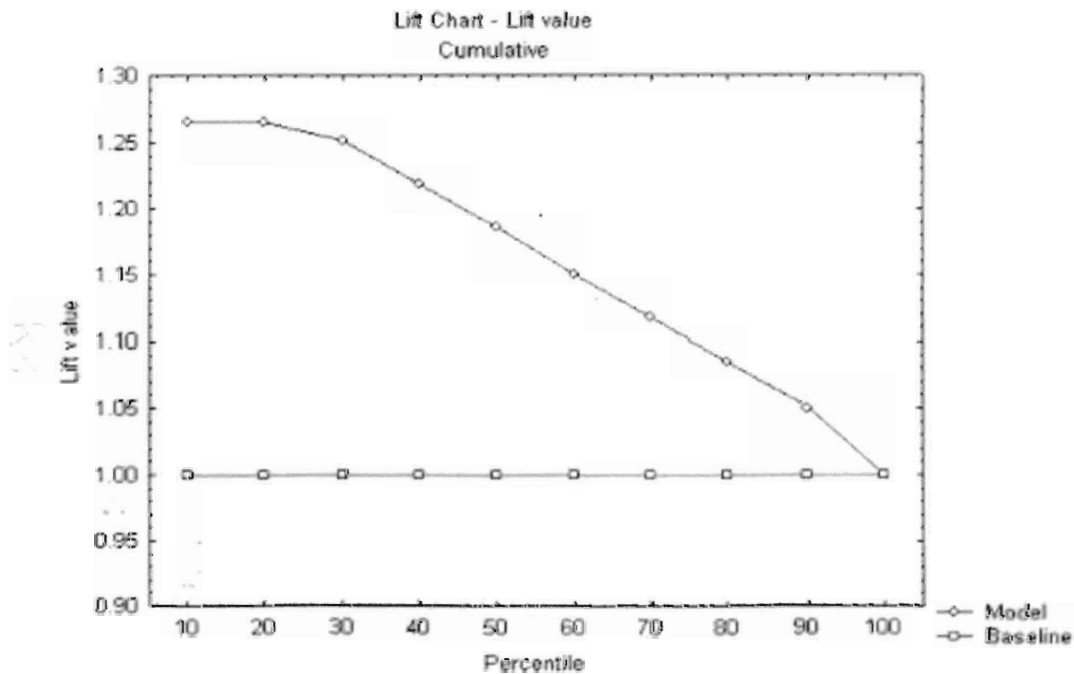


Figure 5: Lift Chart for Possible Injury

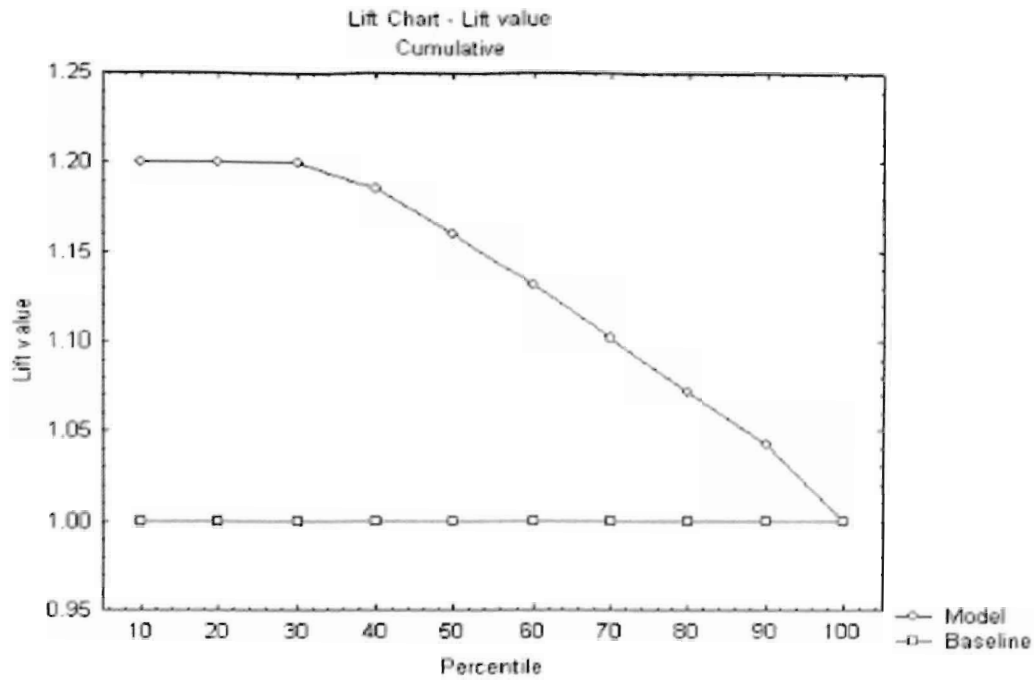


Figure 6: Lift Chart for Non-incapacitating Injury

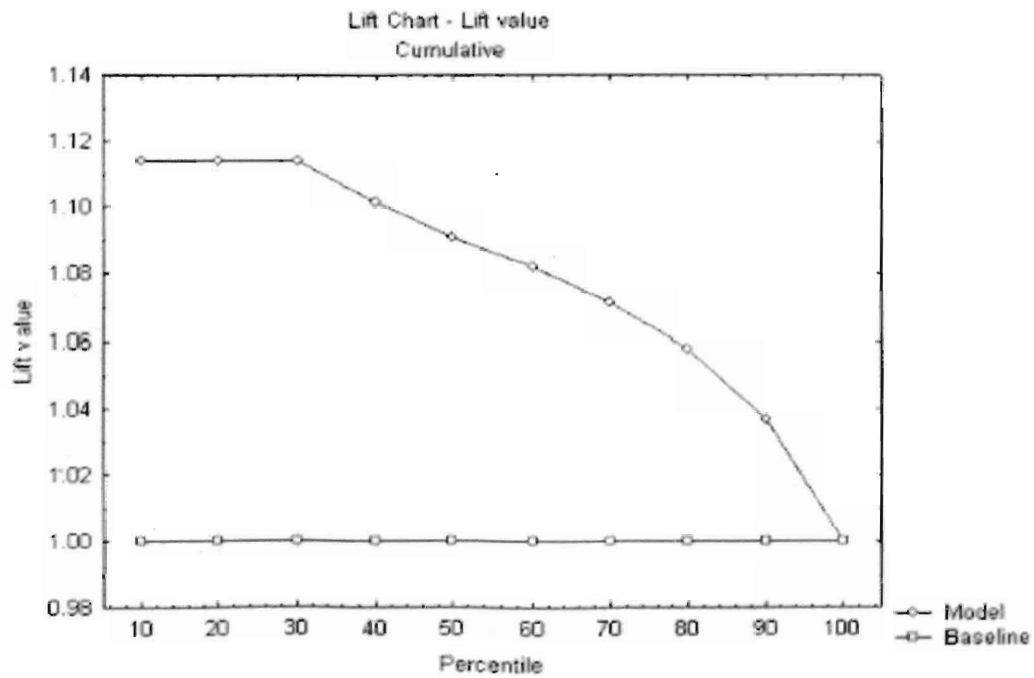


Figure 7: Lift Chart for Incapacitating Injury

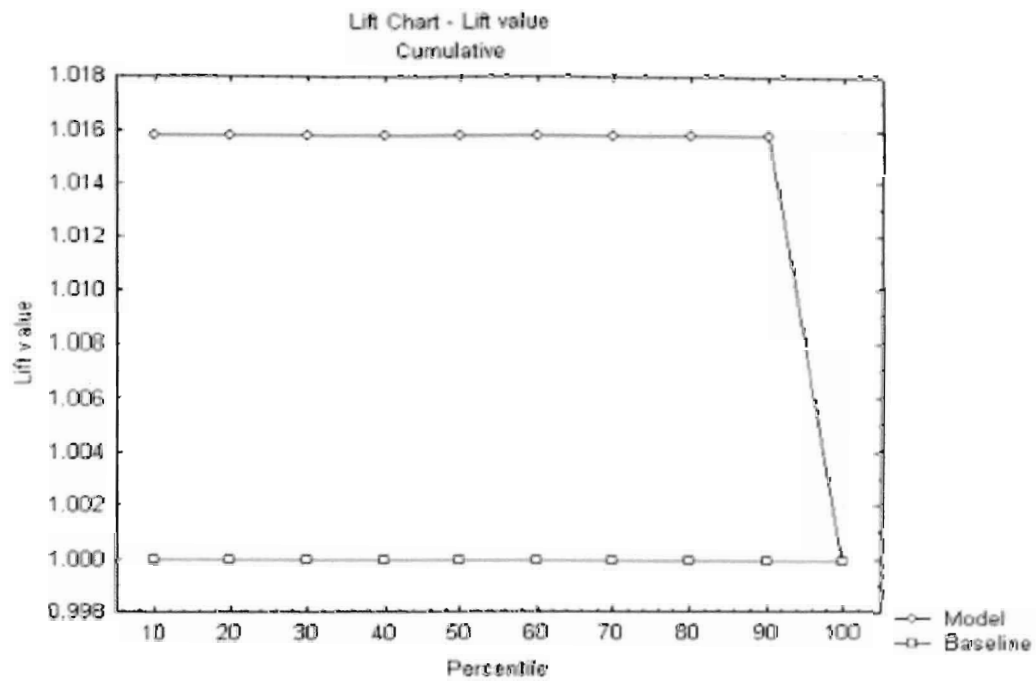


Figure 8: Lift Chart for Fatal Injury

Tree layout for NoInjury
Num. of non-terminal nodes: 355, Num. of terminal nodes: 356

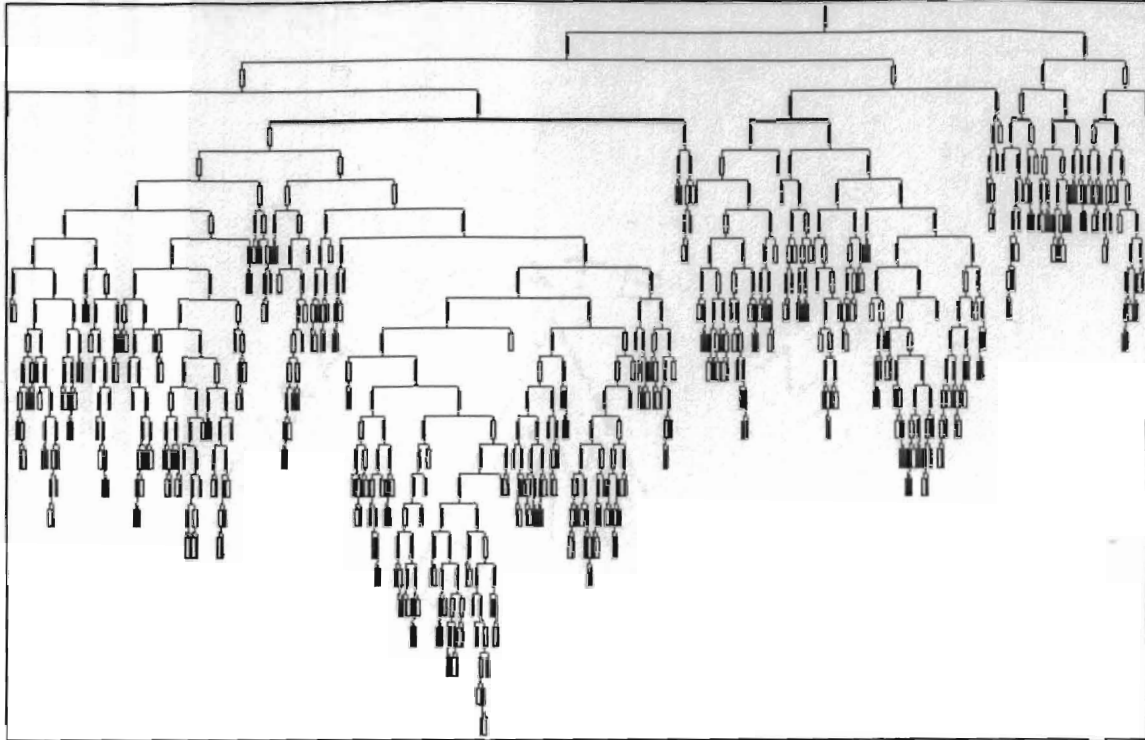


Figure 9: No Injury Tree Structure

Tree layout for PosInjury
Num. of non-terminal nodes: 465, Num. of terminal nodes: 486

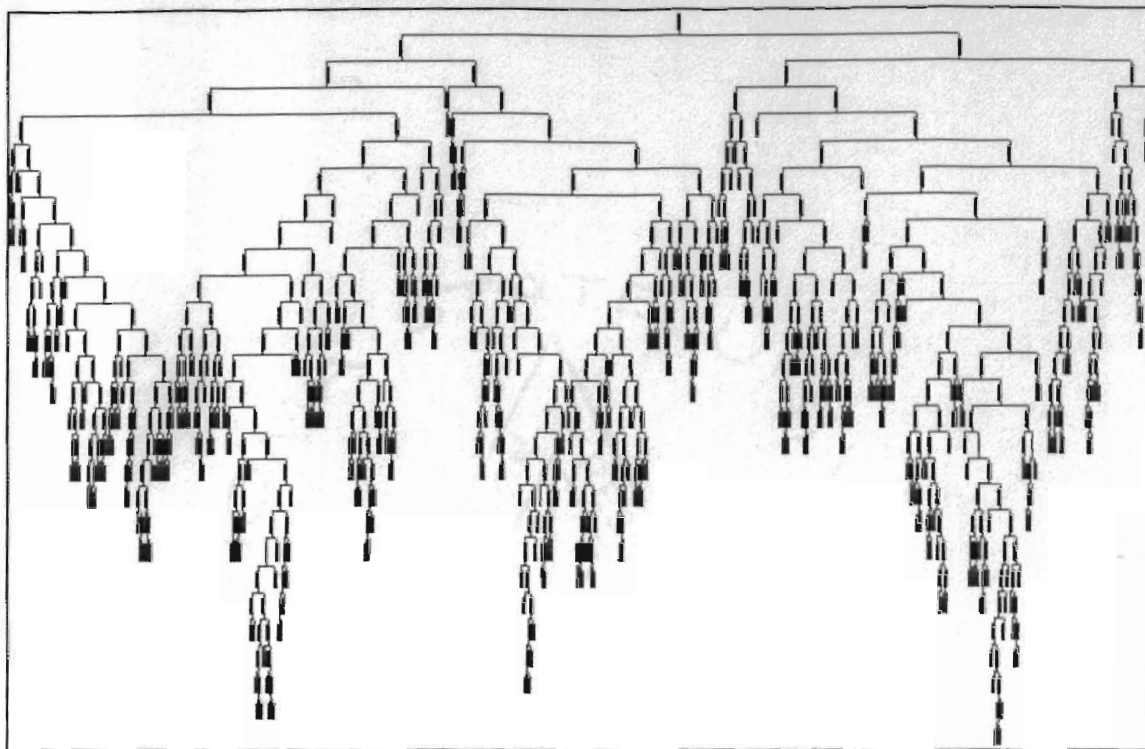


Figure 10: Possible Injury Tree Structure

Tree layout for NonIncap
Num. of non-terminal nodes: 448, Num. of terminal nodes: 449

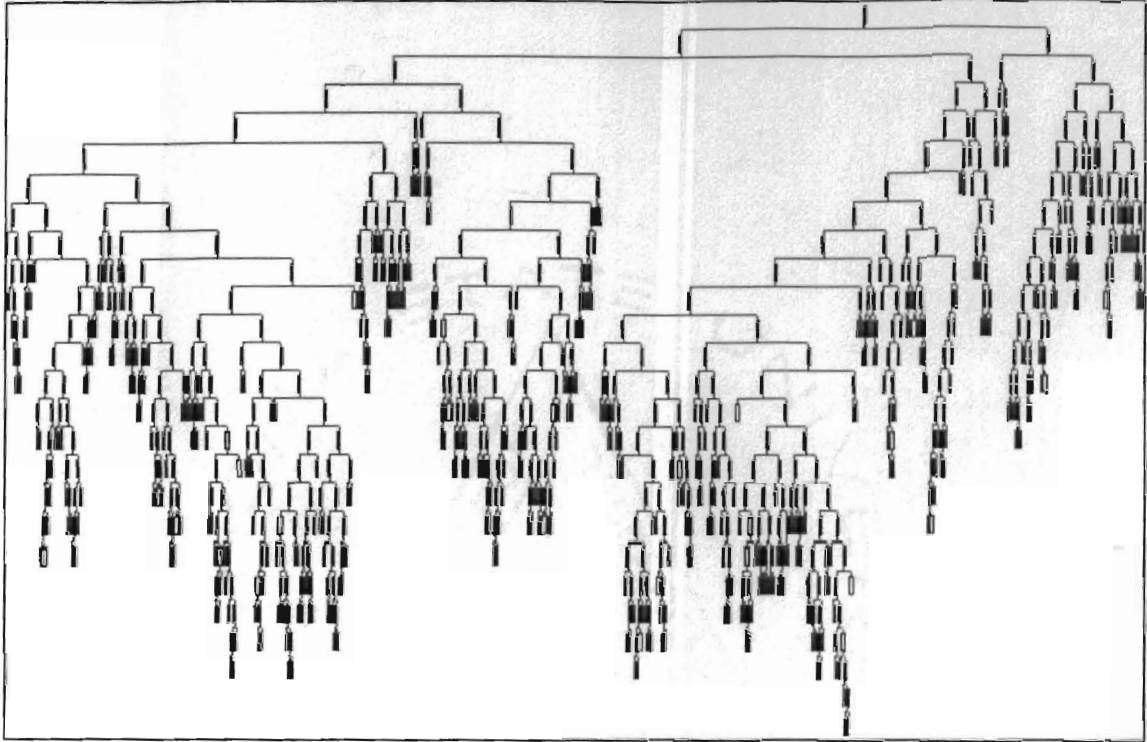


Figure 11: Non-Incapacitating Injury Tree Structure

Tree layout for Incap
Num. of non-terminal nodes: 290, Num. of terminal nodes: 291

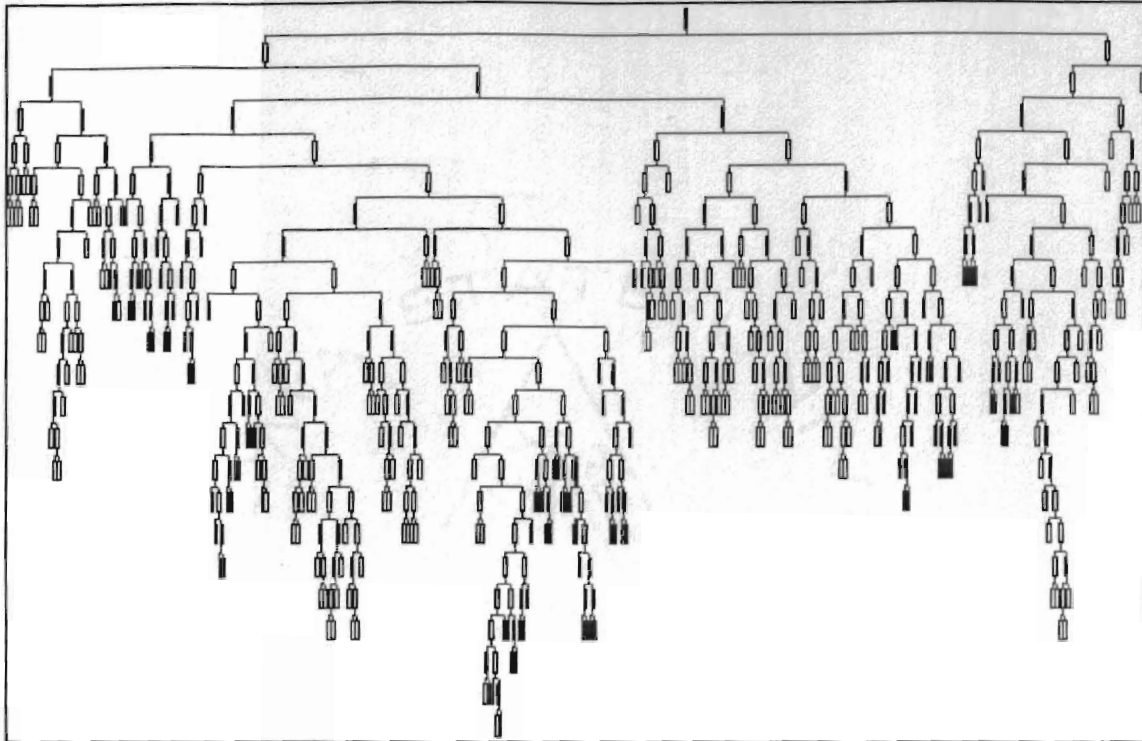


Figure 12: Incapacitating Injury Tree Structure

Tree layout for Fatal
Num. of non-terminal nodes: 149, Num. of terminal nodes: 150

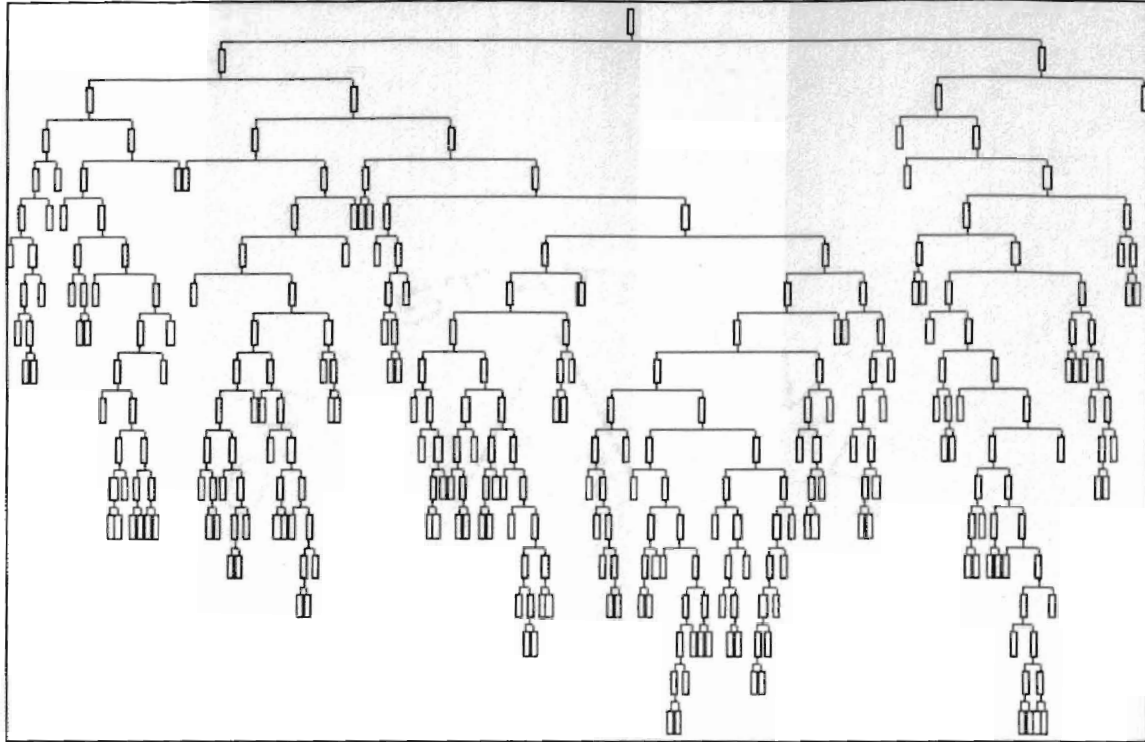


Figure 13: Fatal Injury Tree Structure

5.2.3 SVM Result

SVM^{light} [SVM^{light}] is an implementation of Vapnik's Support Vector Machines (SVMs) [Vapnik] for the problem of pattern recognition, for the problem of regression, and for the problem of learning a ranking function. SVM^{light} was developed by Thorsten Joachims at Cornell University. The program is free for scientific use, it can be downloaded on the website. There are several kernel functions available in SVM^{light}, they are linear kernel, polynomial $(s a * b + c)^d$, radial basis function $\exp(-\gamma \|a-b\|^2)$, sigmoidal $\tanh(s a * b + c)$, and user defined kernel.

Ali et al. compared SVM with Navie Bayes, C4.5, and neural network in terms of accuracy and computational complexity; they found the SVM polynomial kernel is the best choice [Ali]. Before we use SVM^{light}, we need to do data preprocessing to fit the data format that SVM^{light} takes. The first lines may contain comments and are ignored if they start with #. Each of the following lines represents one training example and is in the following format:

```
<line> .= <target> <feature>:<value> <feature>:<value> ... <feature>:<value>
<target> .= +1 | -1 | 0 | <float>
<feature> .= <integer> | "qid"
<value> .= <float>
```

SVM^{light} discards 0 value, our dataset all variable start with 0, for categorical and numerical variables. We convert the 0 to some other number. The numerical variable vehicle age has values range from 0 to 36 that we converted to 1-37. SVMs need separated training and testing dataset, so we divided the dataset into 80% for training, and 20% for testing. The testing dataset was obtained by running the Stratified Random

Sampling algorithm on Webstatistica, extracting 20% of records from each class. Since Webstatistica does not automatically take out the sampling data from the original dataset, the sampling data were deleted from the original data, to obtain the training dataset.

We experimented with polynomial kernel and radial basis function kernel. For some reason, polynomial kernel takes more than ten hours to train one class, so we only focus on radial basis function (RBF). C value is the trade-off between training error and margin, on SVM^{light} the default is $[\text{avg. } x*x]^{-1}$. We first experiment with the default c value, and giving different gamma value, and then we assigned our c value and gamma, but the results of our experiment were not exiting. Table 9 lists the parameters and the accuracy of each experiment for each class.

Table 9: Parameter Setting and Accuracy (%) of RBF SVM

	g=0.0001 default c=42.8758	g=0.001 default c=4.6594	g=0.5 default c=0.5	g=1.2 default c=0.5	g=1.5 c=2	g=2 c=10	g=0.00001 c=100	g=0.0001 c=100	g=0.001 c=100
NoInjury									
class0	59.76	59.80	57.95	57.65	53.62	54.12	57.34	59.76	60.46
class1	60.14	60.14	60.82	55.63	55.73	55.53	62.88	60.14	60.14
overall	59.95	59.95	59.40	56.63	54.69	54.84	60.15	59.95	60.30
PosInjury									
class0	100.00	100.00	100.00	99.88	95.33	95.58	100.00	100.00	100.00
class1	0.00	0.00	0.00	0.00	3.67	3.42	0.00	0.00	0.00
overall	79.70	79.70	79.70	79.60	76.72	76.87	79.70	79.70	79.70
Nonincap									
class0	100.00	100.00	100.00	100.00	97.43	97.49	100.00	100.00	100.00
class1	0.00	0.00	0.00	0.00	3.21	2.92	0.00	0.00	0.00
overall	82.98	82.98	82.98	82.98	81.39	81.39	82.98	82.98	82.98
Incap									
class0	100.00	100.00	100.00	99.89	98.06	98.11	100.00	100.00	100.00
class1	0.00	0.00	0.00	0.00	2.83	2.83	0.00	0.00	0.00
overall	89.48	89.48	89.48	89.38	88.04	88.09	89.48	89.48	89.48
Fatal									
class0	100.00	100.00	100.00	100.00	99.95	99.95	100.00	100.00	100.00
class1	0.00	0.00	0.00	0.00	3.33	3.33	0.00	0.00	0.00
overall	98.51	98.51	98.51	98.51	98.51	98.51	98.51	98.51	98.51

5.3 Model Evaluation

The results of our experiment show that decision tree offers better classification accuracy than neural network and SVMs. We can use one tree for one injury class. For this dataset, SVMs didn't do well; the data is probably too complicated for SVMs to learn. Table 10 shows the overall testing performance comparison of decision tree and neural network. We only compare the testing performance because the testing performance tells how well the model will generalize.

Table 10: Performance Evaluation

Classes	CART (%)	MLP (%)
No Injury	68.16	61.50
Possible Injury	66.28	57.58
Non-incapacitating Injury	66.16	56.80
Incapacitating Injury	72.61	63.43
Fatal Injury	91.61	75.17

CHAPTER VI

SUMMARY

In this thesis project, we studied the automobile accident dataset from 1995 to 2000, and used three machine learning paradigms to predict driver's injury severity in head-on front impact point collisions. From the empirical results, we can see decision tree offers better classification accuracy than neural network and support vector machine. For fatal injury class, decision tree can classify the fatal injury 100%, and classify non-fatal injury 91.47%, the overall predicted accuracy of this model is 91.61%. This model can be used to predict if the accident will cause driver's fatality when an accident happens. The input variables that had the most impact on fatal injury were not using seat belt, light condition, and driver's alcohol usage. This means that all drivers should use seat belt and not drink and drive, when the roadway is dark use extra precautions. This is very important because fatal injury has the highest cost to society economically and socially. Other researchers have found that SVMs can offer more promising results than artificial neural network [Ali, Belousov]. Our dataset is probably too complicated for SVMs to learn. One very important factor of causing different injury levels is the actual speed that the vehicle was going when the accident happened. Our dataset doesn't provide enough information on the actual speed since 67.68% of the data records with an unknown speed. If the speed was available, it might help the models learn better.

SELECTED REFERENCES

- Abdelwahab, H. T. & Abdel-Aty, M. A., Development of Artificial Neural Network Models to Predict Driver Injury Severity in Traffic Accidents at Signalized Intersections. *Transportation Research Record* 1746.
- Abraham, A. & AuYeung, A., Integrating Ensemble of Intelligent Systems for Modeling Stock Indices. Lecture Notes in Computer Science- Volume 2687, Jose Mira and Jose R. Alvarez (Eds.), Springer Verlag, Germany, pp. 774-781, 2003.
- Abraham, A. & Nath, B., Artificial Neural Networks for Intelligent Real Time Power Quality Monitoring System. In Proceedings of *First International Power & Energy Conference*, (INT-PEC'99- Australia), (CD ROM Proceeding), Isreb M (Editor), ISBN 0732 620 945, Australia, 1999.
- Ali, A B M S.& Abraham, A., An Empirical Comparison of Kernel Selection for Support Vector Machines. 2nd International Conference on *Hybrid Intelligent Systems, Soft Computing Systems: Design, Management and Applications*, Abraham A., Köppen M. and Ruiz-del-Solar J. (Eds.), IOS Press, The Netherlands, p p. 3 21-330, 2002.
- Belousov, A. I., Verzhakov, S. A., & Frese, J., A Flexible Classification Approach with Optimal Generalization Performance: Support Vector Machines. *Chemometrics and Intelligent Laboratory Systems*. Vol. 64, 2002, pp.15-25.
- Bigus, J. P., *Data Mining With Neural Networks*. McGraw-Hill, 1996.
- Burges, C. J. C., A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2, 121-167, 1998.
- Campbell, C., Kernel Methods: A Survey of Current Techniques. *Elsevier Science, Neurocomputing*, Vol 48, 2002, pp. 63-84.
- Collobert, R. & Bengio, S., SVM Torch: Support Vector Machine for Large-Scale Regression Problems. *Journal of Machine Learning Research*, Vol 1, pp. 143-160, 2001.
- Cristianini, N. & Shawe-Taylor, J., *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

- Dia, H., & Rose, G., Development and Evaluation of Neural Network Freeway Incident Detection Models Using Field Data. *Transportation Research C*, Vol. 5, No. 5, 1997, pp. 313-331.
- Dunham, M. H., *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2003.
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P., From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, Fall1996, pp. 37-54.
- GES-General Estimates System.
<http://www-nrd.nhsta.dot.gov/departments/nrd-30/ncsa/ges.html>. Access date: October, 2002.
- Han, J., & Kamber, M., *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc, San Francisco, California, 2001.
- Hand, D., Mannila, H., & Smyth, P., *Principles of Data Mining*. The MIT Press, 2001.
- Haykin, S., *Neural Networks: A Comprehensive Foundation*. New York: MacMillan Publishing, 1994.
- Joachims, T., *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002.
- Kim, K., Nitz, L., Richardson, J., & Li, L., Personal and Behavioral Predictions of Automobile Crash and Injury Severity. *Accident Analysis and Prevention*, Vol. 27, No. 4, 1995, pp. 469-481.
- Mussone, L., Ferrari, A., & Oneta, M., An analysis of urban collisions using an artificial intelligence model. *Accident Analysis and Prevention*, Vol. 31, 1999, pp. 705-718.
- Negnevitsky, M., *Artificial Intelligence*. Addison-Wesley, 2002.
- Omar, T., Eskandarian, A. & Bedewi, N., Vehicle Crash Modeling Using Recurrent Neural Networks. *Mathematical Computer Modeling*, Vol. 28, No. 9, pp. 31-42, 1998.
- Pal, M., SVM Application List.
<http://www.clopinet.com/isabelle/Projects/SVM/applist.html>.
- Pyle, D., *Data Preparation for Data Mining*. Morgan Kaufmann Publishers Inc, San Francisco, California, 1999.

- Shankar, V., Mannering, F., & Barfield, W., Statistical Analysis of Accident Severity on Rural Freeways. *Accident Analysis and Prevention*, Vol. 28, No. 3, 1996, pp.391-401.
- Shearer, C., The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, Vol. 5, No. 4, Fall 2000, pp. 13-22.
- Sohn, S. Y., & Lee, S. H., Data Fusion, Ensemble and Clustering to Improve the Classification Accuracy for the Severity of Road Traffic Accidents in Korea. *Safety Science*, Vol. 4, issue1, February 2003, pp. 1-14.
- StatSoft. Electronic Statistics Textbook. <http://www.statsoft.com/textbook/stathome.html>. Access date: October, 2002.
- SVM^{light}. http://www.cs.cornell.edu/People/tj/svm_light/. Access date: May, 2003.
- Vapnik, V. N., *The Nature of Statistical Learning Theory*. Springer, 1995.
- Yang, W.T., Chen, H. C., & Brown, D. B., Detecting Safer Driving Patterns By A Neural Network Approach. *ANNIE '99 for the Proceedings of Smart Engineering System Design Neural Network, Evolutionary Programming, Complex Systems and Data Mining*, Vol. 9, pp 839-844, Nov. 1999.
- Yeo, A. C., Smith, K. A., & Willis, R. J., Clustering Technique for Risk Classification and Prediction of Claim Costs in the Automobile Insurance Industry. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 10, 2001, pp. 39-50.
- Zembowicz, R. & Zytkow, J. M., 1996. From Contingency Tables to Various Forms of Knowledge in Database. *Advances in knowledge Discovery and Data Mining*, editors, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. AAAI Press/The MIT Press, pp.329-349.
- Zurada, J. M., *Introduction to Artificial Neural Systems*. West Publishing Company, 1992.

APPENDICES

APPENDIX A: VARIABLE DEFINITIONS FOR GES DATASET

LABELS

YEAR: The year of the accident.
YYYY Format

MONTH: The month in which the crash occurred.

- 1 = January
- 2 = February
- 3 = March
- 4 = April
- 5 = May
- 6 = June
- 7 = July
- 8 = August
- 9 = September
- 10 = October
- 11 = November
- 12 = December

REGION

- 1 = Northeast (PA, NJ, NY, NH, VT, RI, MA, ME, CT)
- 2 = Midwest (OH, IN, IL, MI, WI, MN, ND, SD, NE, IA, MO, KS)
- 3 = South (MD, DE, DC, WV, VA, KY, TN, NC, SC, GA, FL, AL, MS, LA, AR, OK, TX)
- 4 = West (MT, ID, WA, OR, CA, NV, NM, AZ, UT, CO, WY, AK, HI)

PSU

Primary Sampling Unit: There are 60 possible values ranging from 1 to 97. A PSU is either a large central city, a county surrounding a city, or a group of counties.

PJ

The number (range 1 through 120) of the police jurisdiction from which the PAR was originally sampled.

CASENUM

GES Case Number.

PERNO: Person Number

VEHNO: Vehicle Number

Number assigned to all motor vehicles in transport. Numbers assigned must be consecutive starting with "1" for each crash. (These numbers are computer assigned.)

MAKE: Vehicle Make

A numerical code indicating the make of each motor vehicle in transport. This is only useful for reference. There are too many values here.

MODEL: Vehicle Model

A numerical code indicating the model of each motor vehicle in transport. This is only useful for reference. There are too many values here.

INPUTS

PER_TYPE (NOT USED IN DRIVER ONLY)

1 = Driver

2 = Passenger

SEAT_POS: Seat Position (NOT USED IN DRIVER ONLY)

11 = Front Seat - Left Side (Driver's Side)

12 = Front Seat - Middle

13 = Front Seat - Right Side

18 = Front Seat - Other

19 = Front Seat - Unknown

EJECT

0 = No Ejection

1 = Ejection (Includes Total and Partial Ejection)

AGE

0 = 24 and under

1 = 25- 64

2 = 65 +

SEX

0 = Female

1 = Male

ALCOHOL

Reports alcohol use by a person in the vehicle.

0 = No Alcohol

1 = Alcohol Involved

REST_SYS

Encodes what was documented on the PAR regarding occupant use of available vehicle restraints

(i.e., belts child safety seat, helmet, or automatic restraints).

0 = None Used or Not Applicable

1 = Used (Any Kind)

BODY_TYP: Body Type

0 = Automobiles and Automobile Derivatives

1 = SUVs and Vans

2 = Light and Medium Conventional Trucks

VEH_AGE

Calculated field found by using this formula (YEAR - MODEL_YR + 1)

Values range from 0 to 72

VEH_ROLE: Vehicle Role

Indicates vehicle role in single or multi-vehicle crashes.

0 = Non-Collision

1 = Striking

2 = Struck

3 = Both

IMPACT: Initial Point of Impact

Codes the first impact point that produced property damage or personal injury.

0 = No Damage/Non-Collision

1 = Front

2 = Right Side

3 = Left Side

4 = Back

5 = Front Right Corner

6 = Front Left Corner

7 = Back Right Corner

8 = Back Left Corner

MAN_COL: Manner of Collision

Indicates the orientation of the vehicles in a collision. If a non-collision, it is classified as such.

0 = Not Collision

1 = Rear-End

2 = Head-On

3 = Rear-to-Rear

4 = Angle

5 = Sideswipe, same direction

6 = Sideswipe, opposite direction

ROLLOVER:

Indicates if a rollover occurred (tripped or untripped). Rollover is defined as any vehicle rotation of 90 degrees or more about any true longitudinal or lateral axis. Rollover can occur at any time during the crash.

0 = No rollover

1 = Rollover

VIS_OBSC**SUR_COND: Roadway Surface Condition**

Condition of road surface at the time of the crash.

0 = Dry

1 = Slippery

LGHT_CON: Light Condition

General light conditions at the time of the crash, taking into consideration the existence of external roadway illumination fixtures. (*Note: In 1999 "6" Dawn or Dusk was removed.)

0 = Daylight

1 = Partially Dark

2 = Dark

POSSIBLE INPUTS (too many unknowns)**SPEED: Travel Speed**

Actual miles per hour.

00 = Stopped Vehicle

01-96 = (Actual Travel Speed (MPH))

97 = Ninety-Seven MPH or Greater

99 = Unknown

SPD_LIM: Speed Limit

Actual posted speed limit in miles per hour.

0 = No Statutory Limit (parking lot, alley, etc.)

05-75 = (Actual Speed Limit)

99 = Unknown

OUPUT:**INJ_SEV**

0 = No Injury

1 = Possible Injury

2 = Non-incapacitating Injury

3 = Incapacitating Injury

4 = Fatal Injury

APPENDIX B: SAMPLE DATA FROM ORIGINAL DATASET

YEAR	MONTH	REG	OPS	P	CASE	VEH	PEM	MO	A	S	AL	RES	BO	VEH	MA	ROLL	SUR	LG	SPD	INJ				
					NUM	NO	NOE	DEL	GE	EX	CO	ST	ED	RO	IP	OVER	CON	HT	PE	LIM	SEV			
1997	1	1	1	1	1000001	1	1	49	399	1	0	0	1	0	0	4	1	20	0	1	2	99	99	1
1998	1	1	1	1	1000001	1	1	21	399	1	0	0	1	0	0	11	1	14	0	1	0	99	99	0
1998	1	1	1	1	1000001	2	1	22	399	1	1	0	1	0	0	13	1	14	0	1	0	99	99	0
1997	1	1	1	1	1000002	1	1	21	399	2	1	0	1	0	0	10	2	24	0	1	0	99	99	0
1997	1	1	1	1	1000002	2	1	24	399	1	1	0	1	0	0	2	1	14	0	1	0	99	99	0
1998	1	1	1	1	1000002	2	1	20	499	0	1	0	1	0	2	10	1	34	0	1	1	99	99	0
1997	1	1	1	1	1000003	1	1	22	399	1	0	0	1	0	0	5	1	11	0	1	0	99	99	0
1997	1	1	1	1	1000003	2	1	37	399	1	0	0	1	0	0	2	2	41	0	1	0	0	99	0
1998	1	1	1	1	1000003	1	1	7	499	1	0	0	1	0	2	2	1	10	0	1	1	99	99	0
1997	1	1	1	1	1000004	1	1	12	499	1	0	0	1	0	2	3	1	14	0	1	0	99	99	0
1997	1	1	1	1	1000004	2	1	20	499	1	1	0	1	0	2	12	2	34	0	1	0	99	99	0
1998	1	1	1	1	1000004	1	1	21	399	2	1	0	1	0	0	11	1	14	0	1	1	99	99	0
1998	1	1	1	1	1000004	2	1	14	399	0	1	0	1	0	0	12	1	14	0	1	1	99	99	1
1998	1	1	1	1	1000004	3	1	20	399	1	1	0	1	0	0	9	2	34	0	1	1	0	99	0
1998	1	1	1	1	1000004	4	1	18	399	1	1	0	1	0	0	12	2	14	0	1	1	0	99	0
1997	1	1	1	1	1000005	1	1	20	399	1	0	0	1	0	0	8	1	14	0	1	0	99	99	0
1997	1	1	1	1	1000005	2	1	30	399	0	0	0	1	0	0	4	2	34	0	1	0	99	99	0
1998	1	1	1	1	1000005	1	1	18	399	0	0	0	1	0	0	12	1	14	0	1	1	99	99	0
1998	1	1	1	1	1000005	2	1	49	399	1	0	0	1	0	0	1	2	24	0	1	1	99	99	0
1997	1	1	1	1	1000006	1	1	35	399	1	1	0	1	0	0	11	1	30	0	1	0	99	99	0
1998	1	1	1	1	1000006	1	1	14	399	0	0	0	1	0	0	13	2	24	0	0	1	99	99	0
1998	1	1	1	1	1000006	2	1	34	399	0	1	0	1	0	0	15	1	14	0	0	1	99	99	0
1997	1	1	1	1	1000007	1	1	41	399	0	0	0	1	0	0	5	1	10	0	1	1	99	99	1

APPENDIX C: SAMPLE DATA OF HEAD-ON FRONT IMPACT DATASET

AGE	SEX	ALCHO HOL	REST_S YS	EJECT	BODY_ TYP	VEH_ AGE	VEH_ ROLE	ROLL OVER	SUR_ COND	LGHT_ CON	INJ_ SEV
0	1	0	1	0	0	10	1	0	0	1	0
1	1	0	1	0	0	4	1	0	1	0	2
2	0	0	1	0	0	8	1	0	0	0	3
0	1	0	1	0	0	11	1	0	0	0	2
1	1	0	1	0	2	1	1	0	1	0	1
0	1	0	1	0	0	13	1	0	0	2	1
1	0	0	1	0	0	7	1	0	0	0	0
1	0	0	1	0	0	4	1	0	0	1	1
1	1	0	1	0	1	7	1	0	1	1	0
1	1	0	1	0	0	6	1	0	1	0	2
0	0	0	1	0	0	10	1	0	0	0	2
1	0	0	1	0	0	6	1	0	1	1	2
1	0	0	1	0	2	7	1	0	0	1	2
1	1	0	1	0	0	13	2	0	0	1	0
0	0	0	1	0	0	10	1	0	1	1	3
1	0	0	0	0	2	8	1	0	0	0	2
1	1	0	1	0	2	1	1	0	0	1	4
0	1	0	0	0	0	11	1	0	1	0	3
0	1	0	0	0	0	8	1	0	1	0	2
0	0	0	1	0	0	3	1	0	1	0	0
0	0	0	0	0	0	11	1	0	0	0	3
0	0	0	0	0	0	18	1	0	0	2	4
1	1	0	1	0	1	7	1	0	0	0	2
1	1	0	1	0	0	3	2	0	0	0	0



VITA

MIAO MEI CHONG

Candidate for the Degree of

Master of Science

Thesis: PREDICTING DRIVER'S INJURY SEVERITY IN AUTOMOBILE HEAD-ON COLLISIONS USING MACHINE LEARNING

Major field: Computer Science

Biographical:

Personal Data: Born in Guangzhou, China, November 17, 1963, the daughter of Yu Liu and Hui Ying Chen.

Education: Graduated from Zhixin High School, Guangzhou, Guangdong, China in 1982; received Bachelor of Science degree in Medicine from Guangzhou Medical College in July 1987. Completed the Requirements for the Master of Science degree in Computer Science at Oklahoma State University in August 2003.

Professional Experience: Resident Obstetrics and Gynecology, Guangdong Provincial Health Center for Women & Children, Guangzhou, China, 1987-1991; Teaching Assistant & Research Assistant, Computer Science department, Oklahoma State University, August 2001-May 2003.